

AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application.

LISTING OF CLAIMS

1. (Currently Amended) A method comprising:
issuing an instruction selected from a queue;

enqueueing the instruction issued within a recirculation queue in one of a blocked state and an unblocked state if completion of the instruction is prevented by a detected blocking condition; and

reissuing ~~a selected~~ the instruction from the recirculation queue if a detected blocking condition of at least one instruction within the recirculation queue, other than the ~~selected~~ instruction, is satisfied.

2. (Original) The method of claim 1, wherein issuing comprises:
arbitrating between a plurality of queues to select a queue;
selecting a current instruction from the queue selected; and
issuing the current instruction for the queue selected.

3. (Currently Amended) The method of claim 2, wherein issuing the current instruction comprises:

determining a state of the current instruction;

selecting an alternate queue from the plurality of queues if a state of the ~~selected~~ instruction is blocked; and

issuing an instruction selected from the alternate selected queue.

4. (Original) The method of claim 1, wherein enqueueing comprises:
detecting the blocking condition prohibiting the instruction issued from
completion;

placing the instruction within the recirculation queue;

setting a state of the instruction as blocked to prohibit reissue of the instruction;

and

storing the detected blocking condition.

5. (Original) The method of claim 1, further comprising:
identifying blocking conditions of instructions within the recirculation queue;
determining whether any blocking condition of any instruction within the
recirculation queue is satisfied;

enabling recirculation of instructions from the recirculation queue by setting a
state of each instruction within the recirculation queue to an unblocked state if any
blocking condition is satisfied.

6. (Original) The method of claim 1, wherein reissuing instructions
comprises:

receiving a request to issue an instruction contained within the recirculation
queue;

determining a state of a current instruction of the recirculation queue;

issuing the current instruction if the state of the current instruction is an
unblocked state in response to the received request; and

disregarding the request if the state of the current instruction is a blocked state.

7. (Original) The method of claim 1, wherein enqueueing comprises:
determining whether the detected blocking condition preventing the instruction issued from completion is a transient blocking condition;
setting a state of the instruction to an unblocked state if the detected blocking condition is transient; and
resetting a state of each instruction within the recirculation queue to an unblocked state.

8. (Currently Amended) The method of claim 1, wherein reissuing ~~selected~~the instructions comprises:
issuing an unblocked instruction in response to a received request;
enqueueing the reissued instruction if a blocking condition of the instruction remains unsatisfied;
setting a state of the reissued instruction to a blocked state; and
storing the blocking condition.

9. (Original) The method of claim 1, wherein the detected blocking condition is one of a data blocking condition and a resource blocking condition.

10. (Original) The method of claim 1, wherein the recirculation queue is a first in, first out circular queue.

11. (Currently Amended) An article of manufacture including a machine readable storage medium having stored thereon instructions which may be used to program a system to perform a method, comprising:

issuing an instruction selected from a queue;

enqueueing the instruction issued within a recirculation queue in one of a blocked state and an unblocked state if completion of the instruction is prevented by a detected blocking condition; and

reissuing ~~a selected~~the instructions from the recirculation queue if a detected blocking condition of at least one instruction within the recirculation queue, other than the ~~selected~~-instruction, is satisfied.

12. (Original) The article of manufacture of claim 11, wherein issuing comprises:

arbitrating between a plurality of queues to select a queue;

selecting a current instruction from the queue selected; and

issuing the current instruction from the queue selected.

13. (Currently Amended) The article of manufacture of claim 12, wherein issuing comprises:

determining a state of the current instruction;

selecting an alternate queue from the plurality of queues if a state of the ~~selected~~ instruction is blocked; and

issuing an instruction selected from the alternate selected queue.

14. (Original) The article of manufacture of claim 11, wherein enqueueing comprises:

detecting the blocking condition prohibiting the instruction issued from completion;

placing the instruction within the recirculation queue;

setting a state of the instruction as blocked to prohibit reissue of the instruction;

and

storing the detected blocking condition.

15. (Original) The article of manufacture of claim 11, wherein the method further comprises:

identifying blocking conditions of instructions within the recirculation queue;

determining whether any blocking condition of any instruction within the recirculation queue is satisfied;

enabling reissuing of instructions from the recirculation queue by setting a state of each instruction within the recirculation queue to an unblocked state if any blocking condition is satisfied.

16. (Currently Amended) The article of manufacture of claim 11, wherein reissuing ~~selected~~the instructions comprises:

receiving a request to issue an instruction contained within the recirculation queue;

determining a state of a current instruction of the recirculation queue;

issuing a current instruction if the state of the current instruction is an unblocked state; and

disregarding the request if the state of the current instruction is a blocked state.

17. (Original) The article of manufacture of claim 11, wherein enqueueing comprises:

determining whether the detected blocking condition preventing the instruction issued from completion is a transient blocking condition;

setting a state of the instruction to an unblocked state if the detected blocking condition is transient; and

resetting a state of each instruction within the recirculation queue to an unblocked state.

18. (Currently Amended) The article of manufacture of claim 11, wherein reissuing ~~selected~~ instructions comprises:

issuing an unblocked instruction in response to a received request;

enqueueing the reissued instruction if a blocking condition of the instruction remains unsatisfied;

setting a state of the reissued instruction to a blocked state; and
storing the blocking condition.

19. (Original) The article of manufacture of claim 11, wherein the detected block condition is one of a data blocking condition and a resource blocking condition.

20. (Original) The article of manufacture of claim 12, wherein the recirculation queue is a first in, first out circular queue.

21. (Previously Presented) An apparatus, comprising:
a received instruction queue to store received instructions;
a recirculation queue;
arbitration logic to select one of the received instruction queue and the recirculation queue from which to issue a current instruction; and
blocked instruction detection logic to identify instructions blocked from execution by detected blocking conditions, and to enqueue the instructions onto the recirculation queue in one of a blocked state and an unblocked state, including a respective blocking condition of each instruction within the recirculation queue, wherein instructions having a transient blocking condition are enqueued onto the recirculation queue in the unblocked state.

22. (Original) The apparatus of claim 21, wherein the blocked instruction detect logic further comprises:

blocked condition satisfaction logic to detect whether a blocking condition of an instruction within the recirculation queue is satisfied and to set a state of each instruction within the recirculation queue to an unblocked state if a blocking condition of an instruction within the recirculation queue is satisfied.

23. (Original) The apparatus of claim 21, wherein the arbitration logic to determine a state of a selected instruction, select the received instruction queue if a state of the selected instruction is blocked, and issue an instruction selected from the received instruction queue.

24. (Original) The apparatus of claim 21, wherein the blocked instruction detect logic to determine whether the detected blocking condition is a transient blocking condition, set a state of the instruction placed within the queue to an unblocked state if the detected blocking condition is transient, and reset a state of each instruction within the recirculation queue to an unblocked state to enable reissue of instructions contained within the recirculation queue.

25. (Original) The apparatus of claim 21, wherein the blocked instruction detect logic to enqueue a reissued instruction if a blocking condition of the instruction remains unsatisfied, to set a state of the reissued instruction to a blocked state and to store the blocking condition.

26. (Previously Presented) A system comprising:
a memory controller coupled to a memory;

a processor coupled to the memory via a bus, the processor including:

a bus interface unit coupling an execution core to a cache memory

including:

a received instruction queue to store received instructions,

a recirculation queue,

arbitration logic to select one of the received instruction queue and the recirculation queue from which to issue a current instruction, and

blocked instruction detection logic to identify instructions blocked from execution by detected blocking conditions, and to enqueue the instructions onto the recirculation queue in one of a blocked state and an unblocked state, including a respective blocking condition of each instruction within the recirculation queue, wherein instructions having a transient blocking condition are enqueued onto the recirculation queue in the unblocked state.

27. (Original) The system of claim 26, wherein the blocked instruction detect logic further comprises:

blocked condition satisfaction logic to detect whether a blocking condition of an instruction within the recirculation queue is satisfied and to set a state of each instruction within the recirculation queue to an unblocked state if a blocking condition of an instruction within the recirculation queue is satisfied.

28. (Original) The system of claim 26, wherein the arbitration logic to determine a state of a selected instruction, select the received instruction queues if a

state of the selected instruction is blocked, and issue an instruction selected from the received instruction queue.

29. (Original) The system of claim 26, wherein the blocked instruction detect logic to determine whether the detected blocking condition is a transient blocking condition, set a state of the instruction placed within the queue to an unblocked state if the detected blocking condition is transient, and reset a state of each instruction within the recirculation queue to an unblocked state to enable reissue of instructions contained within the recirculation queue.

30. (Original) The system of claim 26, wherein the blocked instruction detect logic to enqueue the reissued instruction if a blocking condition of the instruction remains unsatisfied, to set a state of the reissued instruction to a blocked state and to store the blocking condition.

31. (Currently Amended) ~~[[The]]~~A method comprising:
issuing an instruction selected from a queue;
enqueueing the instruction issued within a recirculation queue in one of a blocked state and an unblocked state if completion of the instruction is prevented by a detected blocking condition;
resetting a state of ~~at least one selected~~the instruction within the recirculation queue if a detected blocking condition of at least one instruction within the recirculation queue, other than the ~~at least one selected~~ instruction, is satisfied; and

reissuing ~~a current~~the instruction from the recirculation queue if a state of the ~~current~~ instruction is indicated as the unblocked state.

32. (New) A method comprising:

issuing a first instruction from a queue;

detecting a first blocking condition for the first instruction prior to execution of the first instruction;

setting the first instruction to one of a blocked state and an unblocked state based on the first blocking instruction;

enqueueing the first instruction within a recirculation queue in one of the blocked state and the unblocked state if completion of the instruction is prevented by the first blocking condition; and

reissuing the first instruction from the recirculation queue if the first blocking condition is satisfied.

33. (New) The method of claim 32 further comprising:

detecting a second blocking condition for a second instruction, wherein the second blocking condition differs from the first blocking condition and the second instruction differs from the first instruction; and

reissuing the first instruction from the recirculation queue if the second blocking condition is satisfied.

34. (New) The method of claim 32 further comprising:
setting the first instruction to an unblocked state based on the first blocking condition; and

enqueueing the first instruction within the recirculation queue in the unblocked state until the first blocking condition is satisfied.

35. (New) The method of claim 34 further comprising:
detecting a second blocking condition for a second instruction, wherein the second blocking condition differs from the first blocking condition and the second instruction differs from the first instruction;

setting the second instruction to a blocked state based on the second blocking condition; and

enqueueing the second instruction within the recirculation queue in the blocked state until the second blocking condition is satisfied.

36. (New) The method of claim 32 wherein enqueueing comprises:
determining whether the first blocking condition is a transient blocking condition; and

setting the first instruction to the unblocked state if the first blocking condition is transient.